

CSS

Estamos muy complacidos de haber compartido contigo este travesía a través de CSS. Esperamos que hayas disfrutado de las actividades y recursos relacionados a este contenido y hayas podido alcanzar los logros anticipados para este curso.

Durante este espacio de aprendizaje nos enfocamos en cuatro aspectos que dividimos en cuatro grandes bloques didácticos: Introducción a CSS, Propiedades de CSS, Diseño Adaptativo y Interacciones en CSS.

HTML es un lenguaje que revolucionó la Internet porque permitió el despliegue de contenidos dispersos en diversos servidores localizados en cualquier lugar del mundo fácilmente, a través del uso de un conjunto de etiquetas. Sin embargo, esto no fue suficiente para los usuarios, quienes pronto comenzaron a demandar mejores presentaciones de estos contenidos. **CSS** es la respuesta a estas demandas para otorgar **formato a los contenidos HTML**. CSS hace uso de un conjunto de **reglas**, que incluyen **selectores**, **propiedades** y **valores**, las cuales examinamos cuidadosamente en este curso.

La segunda parada en nuestro espacio de aprendizaje es en las propiedades de CSS, que es una exploración al **modelo de cajas** de este lenguaje y algunas propiedades que nos permiten operar sobre estas cajas. En esta misma estación nos volcamos a las **tipografías** que constituyen el corazón de los textos de nuestros contenidos. Existe un gran número de propiedades que facilitan el proporcionar estilos a nuestras letras para adaptarlas a nuestros sitios Web.

Es imposible no hablar de diseño adaptativo hoy en día donde existe una amplia gama de dispositivos con diferentes características y tamaños de pantallas. El **diseño adaptativo** es la base para acomodar los contenidos HTML a cada dispositivo, de forma tal de que el usuario se sienta cómodo visualizando los mismos, sin importar qué equipo está empleando. Los **media queries**, los **media types** y el **viewport** son conceptos esenciales de CSS que nos permiten realizar un diseño adaptativo. Finalmente, en esta parada analizamos diversos **Patrones de Diseño Web Adaptativo: Tiny Tweaks, Mostly Fluid, Column Drop, Layout Shifter y OffCanvas**.

Para culminar con esta aventura a través de CSS, estudiamos las transformaciones en CSS, que nos permiten aplicar diversos cambios en los

elementos HTML. Ligado a esto también encontramos las **animaciones**, que dan vida a nuestros contenidos, a través del uso de las **transiciones** y **@keyframes**. Otro aspecto de las transformaciones son los **filtros CSS** usados para la aplicación de efectos espaciales, en particular a las imágenes. La parte final de nuestro estudio son las características avanzadas de CSS que incluyen las funciones y grids.

A través del uso de un enfoque teórico-práctico, has ido obteniendo las habilidades necesarias para complementar tus contenidos HTML, otorgándole estilos que le permitirán a los usuarios disfrutar de ellos desde la comida de cualquier dispositivo.

Luego de haber culminado exitosamente el curso de CSS, has desarrollado las siguientes capacidades:

- Identificar la estructura de una Hoja de Estilo en Cascada (CSS) para proporcionar formato a los contenidos Web.
- Aplicar los componentes estructurales de la Hoja de Estilo en Cascada (CSS) en la construcción de interfaces Web respetando las mejores prácticas en los desarrollos Web Frontend.
- Crear sitios Web usando la Hoja de Estilo en Cascada (CSS) para la producción de interfaces visuales atractivas y adaptativas (responsive).

UNIDAD1: INTRODUCCIÓN A CSS

La **Hoja de Estilo en Cascada** o **CSS** por sus siglas en inglés (**Cascading Style Sheets**) es un mecanismo que sirve para adicionar estilos a un documento Web con el fin de modificar su presentación. Al diseñar una página web empleamos el **HTML** para definir la estructura del documento y **CSS** para aplicar el formato gráfico.

Un recurso interesante para los desarrolladores Web son los **validadores**, que son herramientas que permiten **verificar que el código HTML o CSS** sea correcto, es decir, que cumpla con los lineamientos y recomendaciones de la Word Wide Web Consortium (W3C). La W3C provee una herramienta para realizar las validaciones de HTML y CSS: <http://validator.w3.org/>.

A continuación un resumen de **cómo funciona CSS**. Este mecanismo consiste de reglas y cada regla está formada por un **selector** y una **declaración**, cuyo significado y estructura se muestra en la **Figura 1**.

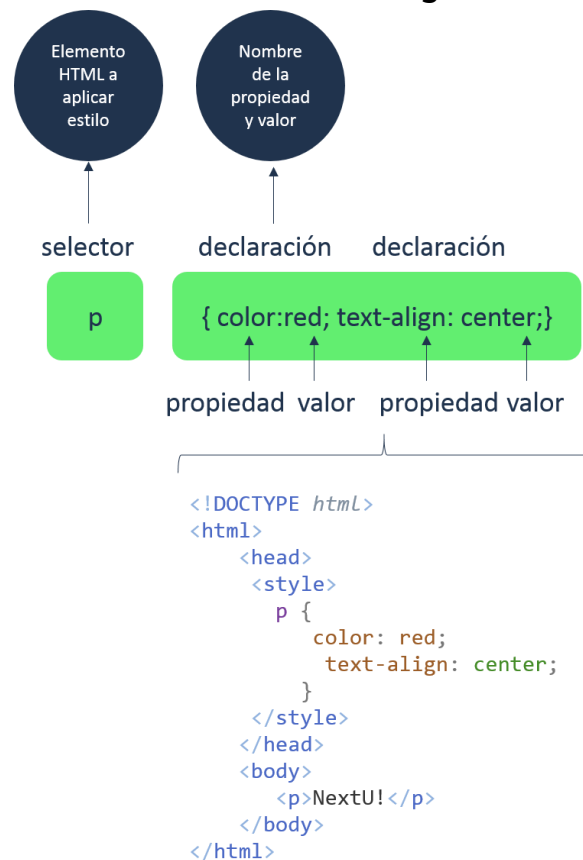


Figura 1.1: Sintaxis de CSS.

[imágenes/WEB19S_C2_Cierre.ppt -Fig 1.1](#)

Hay tres formas de **insertar los estilos CSS** en un documento HTML:

- **Forma interna:** se logra a través del uso de la etiqueta `<style>` ubicándola en el encabezado (header) de una página HTML. Los estilos aplican a la página HTML.
- **Forma en línea:** se logra a través del uso del atributo **style** y este estilo solo aplica al elemento al que está asociado el atributo.
- **Forma externa:** se consigue invocando a un archivo de estilos CSS externo empleando la etiqueta `<link>`, que debe estar en el sección del encabezado (header). Esta forma se puede usar para proporcionar estilo a todo el sitio Web.

Adicionalmente, hay **cinco tipos de selectores**:

- **Selector de clase:** este selector se identifica con el prefijo **punto (.)** y se aplica en conjunción con el atributo `class`. Sirve para aplicar un mismo estilo a muchos elementos.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .ejemplo {
        color: blue;
      }</style>
    </head>
    <p class="ejemplo">Ejemplo</p>
  </html>
```

- **Selector de ID:** se emplea incorporando el **atributo ID** al elemento al que deseamos aplicarle el estilo. La sintaxis del selector usa el símbolo **numeral #**. Se utiliza para aplicar estilo a un único elemento de la página.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo</title>
    <style>
      #ejemplo{
        color: brown;
      }
    </style>
  </head>
  <body>
    <p id="ejemplo">Párrafo 2</p>
  </body>
</html>
```

- **Selector de etiqueta:** el estilo se le aplica a todos los elementos que tengan la etiqueta en el selector.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo</title>
    <style>
      p {
        color: yellow;
      }
    </style>
  </head>
  <body>
    <p>Párrafo 2</p>
  </body>
</html>
```

- **Selector universal:** se emplea para aplicar estilo a todos los elementos de una página. Para ello se usa el símbolo **asterisco** *.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo</title>
    <style>
      * {
        color: yellow;
      }
    </style>
  </head>
  <body>
    <h1> Encabezado</h1>
    <p>Párrafo 2</p>
  </body>
</html>
```

- **Selector descendente:** se emplea para aplicar estilo a los elementos incluidos dentro de otros elementos. Un elemento se considera que está dentro de otro, si se encuentra entre las etiquetas de apertura y cierre de ese elemento.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo</title>
    <style>
      p strong {
        color: brown;
      }
    </style>
  </head>
  <body>
    <p>Selector <strong>descendente</strong></p>
  </body>
</html>
```

Podemos encontrar muchos estilos en una página, es por ello que necesitamos establecer un orden para su aplicación. En CSS, los estilos “cascadean” de acuerdo a las siguientes prioridades, donde el número 1 tiene la más alta prioridad:

1. Estilos en línea.
2. Estilos externos e internos.
3. Estilo del navegador (CSS predeterminado).

Cuando los estilos son del mismo tipo, ellos se leen de acuerdo al orden que han sido definidos, comenzando por el primer estilo definido hasta el último. El último estilo tiene más prioridad que la definición anterior y así sucesivamente (ver ejemplo en **Figura 1.2**).

CSS también permite combinaciones de propiedades dentro de los estilos (ver ejemplo en **Figura 1.3**). Por otra parte, si deseamos romper con esta forma de aplicación en cascada, podemos usar la directiva **!important** junto con el estilo para incrementar la prioridad de ese estilo sobre las otras definiciones.

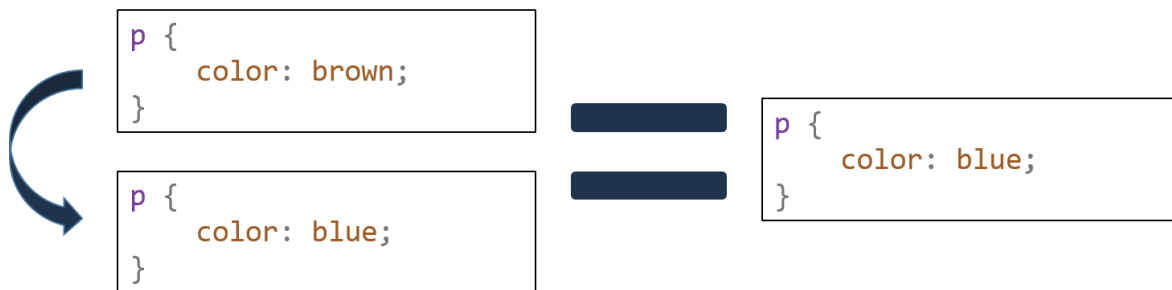


Figura 1.2: Ejemplo de aplicación de estilos en cascada en CSS.

[imágenes/WEB19S_C2_Cierre.ppt -Fig 1.2](#)

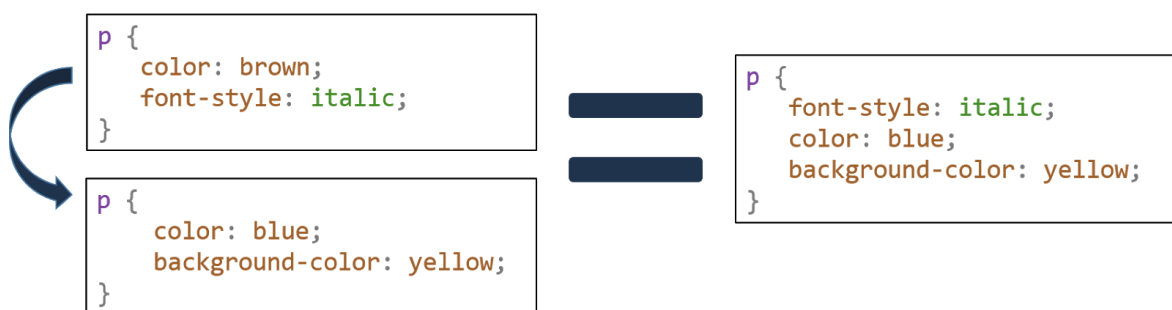


Figura 1.3: Un ejemplo más complejo de la aplicación de estilos en cascada en CSS.

[imágenes/WEB19S_C2_Cierre.ppt -Fig 1.3](#)

UNIDAD 2: PROPIEDADES CSS

Uno de los conceptos fundamentales del lenguaje CSS es el **modelo de caja (Box Model)**, debido a que todos los elementos HTML son tratados como si fueran cajas que se organizan jerárquicamente unas dentro de otras, siendo la caja principal la que corresponde a la etiqueta `<html>` del documento web. la características de la caja se muestran en la Figura 2.1.

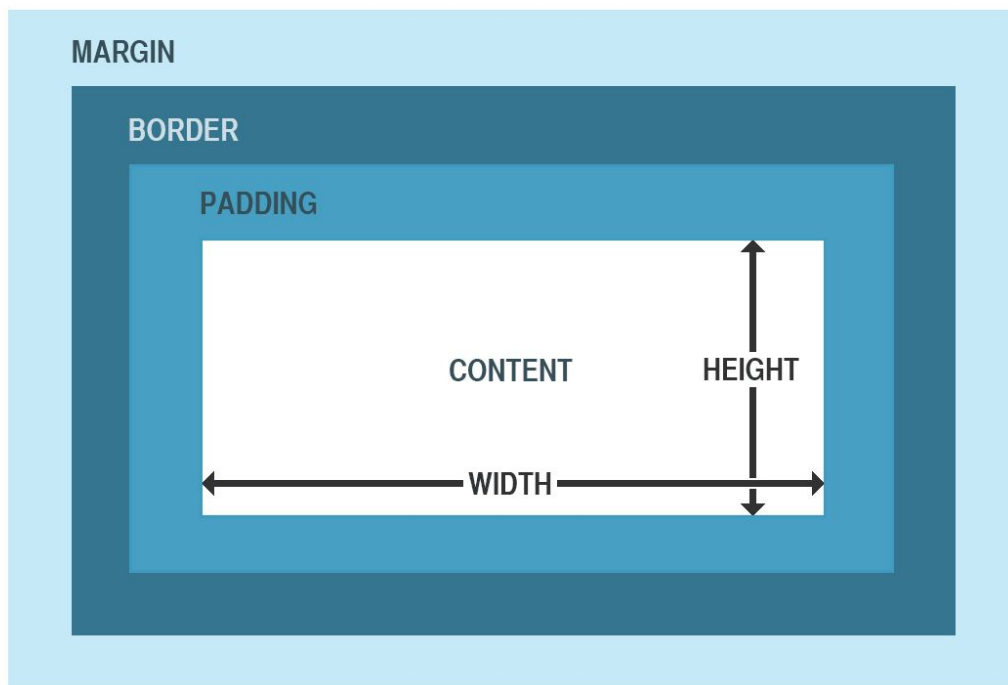


Figura 2.1: Modelo de Caja.

Ligados a una gran cantidad de propiedades de CSS están las unidades de medidas, que se dividen en:

- **Unidades absolutas:** están bien definidas y entre ellas encontramos cm y mm.

- **Unidades relativas:** dependen de otros valores de referencia para adaptarse, tal como el porcentaje (%).
- **Relativas a la tipografía:** entre ellas encontramos: **em** la cual está basada en el elemento que contenga al elemento; **rem** que se basa en el tamaño de la fuente que tenga la raíz del documento HTML es decir el `< body >`; y **ex** que se calcula en función de la altura de la letra “x” minúscula del tipo y del tamaño de la letra del elemento padre.
- **Relativas al viewport o área visible del navegador/ventana:** que incluyen:, **vw** (Viewport Width) que es relativo al ancho del **viewport** y **vh** (Viewport Height) que es relativo al alto del **viewport**.

Asociado al **modelo de cajas** encontramos las siguientes propiedades:

- **Display:** especifica como una caja es desplegada, tal como en línea, en bloque o que no se despliegue.
- **Float:** posiciona y formatea el contenido especificando si un elemento debe aparecer a la izquierda o a la derecha de su contenedor o no “flotar” en lo absoluto.
- **Position:** especifica el método de posicionamiento de una caja, tal como relativo, fijo, y absoluto.
- **Flex:** permite alterar las dimensiones de un elemento y llenar el espacio disponible.

Otro aspecto importante estudiado en esta unidad está relacionado al concepto de **diseño fluido**, que se basa exclusivamente en tener contenedores y elementos con anchos en porcentajes o en unidades de medidas relativas al **viewport** (VH y VW).

El texto de una página Web debe seguir un **tipografía** determinada, la cual especifica el tipo de letra a usar. Aunque existen muchos estilos de letras en el mercado, no todas ellas son de libre uso y no están soportadas por todos los navegadores. El uso de un estilo de letra es importante en el despliegue de nuestros contenidos de forma adecuada en las plataformas existentes. Existen unos tipos de tipografía “seguras” que son estándares en diversos sistemas, Windows y Mac OS, tales como: **Arial o Helvetica, Courier o Courier New, MS Sans Serif o Geneva, MS serif o New York y Times New Roman o Times**.

CSS ofrece un gran número de propiedades para cambiar el formato y proporcionar estilo al texto de nuestro contenido HTML.

UNIDAD 3: CSS DISEÑO ADAPTATIVO

El **Diseño Web Adaptativo** es una técnica que permite desarrollar páginas web que se adapten a cualquier dispositivo. Consiste en explotar las características de HTML y CSS3 con el fin de acomodar los contenidos de una página para que se vean bien en cualquier tipo de plataforma.

Para lograr que un proyecto Web sea adaptativo, debemos determinar el viewport, o el área de la página web que será visible para el usuario, y definir un **conjunto de condiciones (media queries)** y reglas CSS3 de acuerdo con cada tamaño de pantalla, esto, estableciendo las características de los elementos para que puedan verse correctamente en cada dispositivo.

El **viewport** se configura dentro de la etiqueta `<meta>` a través de un elemento. Entre los valores de configuración que podemos especificarle se encuentran:

- Nivel de escalado que puede realizar el usuario.
- El zoom inicial que debe realizar el navegador.
- El área disponible para renderizar.

Los **media queries** son una extensión de CSS que permiten detectar las características de los dispositivos que accedan a nuestra Web y a partir de esos datos crear hojas de estilo más personalizadas, dependiendo de las condiciones que el navegador encuentre en el dispositivo. Las condiciones se dan a través de los **media types**, entre los cuales encontramos: **print** - vista de Impresión, **min-width** - tamaño mínimo en horizontal, **max-width** - tamaño máximo en horizontal, **min-height** - tamaño mínimo en vertical y **max-height** - tamaño máximo en vertical.

Por otro lado, tenemos los **Patrones de Diseño Web** que permiten definir la características de una página Web, tanto lo que corresponde con la distribución de sus elementos, como su adaptabilidad a diversos dispositivos. Entre los patrones que exploramos están:

- **Tiny Tweaks:** es un patrón sencillo que se basa en el uso de una sola columna.
- **Mostly Fluid:** se basa en el uso de una grilla con múltiples filas y columnas. Las columnas se apilan una encima de la otra a medida de que el tamaño de la pantalla se haga más pequeño. Para tamaños más grandes los bloques se acomodan y se añaden espacios y márgenes.
- **Column Drop:** permite definir columnas primarias y secundarias en la página para acomodarlas prioritariamente a medida que se reduce el tamaño de la pantalla.
- **Layout Shifter:** cambia completamente el patrón de los elementos de una página a medida que cambian las dimensiones de la pantalla donde se despliega el contenido.
- **Off Canvas:** en tamaños de pantalla pequeños, este patrón sólo deja en pantalla un elemento y permite el acceso a los demás mediante gestos o botones, desplazando lateralmente el contenido actual.

UNIDAD 4: CSS INTERACCIONES

La **propiedad de transformación** permite realizar ciertos cambios en 2D y 3D a un elemento e incluye:

- **Rotate ():** permite rotar un elemento.
- **Scale():** cambia el tamaño de un elemento en el plano 2D.
- **Perspective ():** permite especificar qué tan lejos está un objeto del usuario.
- **Translate():** relocaliza un elemento en el eje horizontal o vertical.
- **Skew():** sesgar un elemento en el eje horizontal un determinado ángulo. property defines how far the object is away from the user.

Para incluir animaciones en los contenidos tenemos varias opciones. La primera es usar la **propiedad transition** que permite cambiar los valores de las propiedades lentamente de acuerdo al valor introducido. Por otra parte, tenemos la regla **@keyframes** que permite especificar cómo se van cambiando los estilos aplicados a un elemento gradualmente.

CSS nos ofrece la posibilidad de aplicar filtros a los elementos. Los **filtros** no son más que efectos especiales, tales como la saturación y difuminado aplicados a través de la **propiedad filter**.

La última parte de nuestro espacio de aprendizaje de CSS la dedicamos a examinar ciertas características avanzadas de CSS. Una de estas características es el uso de funciones bien definidas, que pueden ser incorporadas en las propiedades para facilitar el establecimiento de formatos. Las funciones soportadas hasta ahora son:

- **rgb()**: define un color en RGB.
- **rgba()**: define un color en RGB con transparencia.
- **hsl()**: define un color en HSL (Hue-saturation-lightness).
- **hsla()**: define un color en HSL con transparencia.
- **radial-gradient()**: establece un gradiente radial en el fondo de la imagen
- **linear-gradient()**: establece un gradiente lineal en el fondo de la imagen.
- **calc()**: permite realizar operaciones matemáticas cuyos resultados se usan en la propiedades de CSS.
- **attr()**: devuelve el valor de un atributo de elemento seleccionado.
- **var()**: permite el uso de variables con valores específicos que pueden ser usadas varias veces en el documento HTML.

Por otra parte, tenemos **las grillas** que permiten definir rejillas usadas para acomodar los elementos de un página de una forma flexible.